

调度自动化机房网络配线管理软件设计与实现

李伟青, 陈龙成, 梅咏武

(广东电网有限责任公司梅州供电局, 广东 梅州 514021)

摘要: 针对公司调度自动化机房网络配线无图纸管理和标签标识不规范的情况, 自主研发了一套机房网络配线管理软件; 首先, 分析了配线可视化管理软件应对网络配线绘图和标签信息自动生成的功能需求, 并运用 MVC 模式理论进行软件架构设计; 其次, 重点阐述了基于 QT 图形库的关键功能实现, 包括核心图元类的构建、元件库组件可配置生成、图形序列化反序列化操作、拓扑查找、设备图元属性管理及标签信息自动生成功能等; 最后, 通过举例说明该软件在实际机房管理中的使用方法和应用成效; 应用结果表明, 通过绘制整个机房的网络连接图并录入相关设备属性信息, 有助于提高机房网络故障的诊断效率和降低机房系统的运维风险。

关键词: 机房; 网络配线; 软件设计; 绘图; 标签

Design and Implementation of Dispatching Automation Room's Network Wiring Management Software

Li Weiqing, Chen Longcheng, Mei Yongwu

(Guangdong Power Grid Co., Ltd. Meizhou Power Supply Bureau, Meizhou 514021, China)

Abstract: For the company dispatch automation room's wirings hadn't drawing management and the labels were not standard, a set of computer room network wiring management software was developed independently. Firstly, the functional requirements of visual wiring management software to deal with drawing network wiring and generating label information automatically were analyzed, and the software architecture based on MVC model theory was designed; Secondly, the implementation of the key functions based on QT graphics library platform were elaborated, including the construction of the graphic class, the generation of components library that can be configured, the graph serialization and anti-serialization operation, topology search, the attribute management of components and the automatic generation of label information; Finally, the actual usage and application effect of the software was illustrated through an example. The results of application show that drawing the entire room network connection diagram and input the device attribute information help to improve the efficiency of network fault diagnosis and reduce the risk of system operation and maintenance.

Keywords: computer room; network wiring; software design; drawing; label

0 引言

电力调度自动化系统主站机房是服务器、工作站、交换机、路由器等网络设备密集分布的场所。自动化系统通常采用星型的网络结构, 因此主干交换机屏柜汇聚了大量设备的网络接入。为使布线整洁美观, 设备机柜的网线往往捆绑成股, 导致难以厘清网线与设备端口之间的对应关系。另外, 不同屏柜之间的业务交互, 需要通过机柜配线架跳线转接来实现, 由此产生的众多网线标签依赖于人工手动编辑打印, 难以保证标签描述的准确性和规范性。在网络接线无图纸参照, 且标签不清晰或者不正确的情况下, 一旦发生网络链路故障, 管理人员需顺着端口接线逐一查找, 不仅排查效率低, 还容易因误插拔正常运行网络, 造成业务被中断。

基于以上事实, 我们自主研发了一套机房网络配线可视化管理软件系统, 通过采用面向对象思想进行架构设计^[1], 可实现机房网线连接关系图绘制和展示, 端口快速查找高亮拓扑, 标签按标准模板自动生成等功能。可视化管理系统实现了机房

网络连接关系的数字化、图纸化, 为网络链路故障诊断提供快速、准确的依据。原来依赖手动编辑制作的设备标签, 现可通过管理系统关联设备属性信息自动生成, 杜绝了标签标识内容不准确、格式不规范的情况。

1 软件总体设计

1.1 功能需求分析

在网络配线管理软件设计之前需对复杂的机房配线连接关系和机房管理过程进行分析, 具体功能需求如下:

1) 将机房设备之间的网络连接关系可视化、图纸化。机房设备包括服务器、工作站, 集线器, 交换机, 配线架、网线等。通过基本的分析, 可以将设备大体划归为两类: 一类是带有含有一定数量网口的设备图元; 另外一类是连通端口关系的连线图元。软件需具备绘图平台且图元可定制, 提供图元连接关系的交互式操作、设备信息录入功能。

2) 能辅助调度自动化系统运维人员管理机房配线。机房配线管理的内容包括: 对不同类型设备根据业务功能分类、编号、贴标签, 通过名称或者 IP 快速检索到设备及设备之间的连接关系, 高亮显示指定网络连接路径以便故障排查等。针对这些业务需要, 给基本的两类图元进行添加属性绑定功能。将基本的管理功能数字化成对设备属性的过滤与检索, 从而实现

收稿日期: 2015-09-06; 修回日期: 2015-09-27。

作者简介: 李伟青(1986-), 男, 广东梅州人, 硕士, 工程师, 主要从事应用软件开发, 电力调度自动化系统维护方向的研究。

复杂的业务逻辑。

- 综上所述, 软件系统需要提供如下的功能:
- 1) 基本的图形绘制的功能, 包括配置基本的机房图元, 绘制图元之间的连接关系;
 - 2) 提供基本的图元与图形编辑功能, 包括画布的大小配置, 图元的移动缩放, 图元属性的录入与修改等;
 - 3) 图形的持久化保存, 载入重现, 图纸打印等文件管理;
 - 4) 高阶应用功能, 如连接关系的拓扑查找, 设备标签按照指定格式生成并导出。

1.2 系统的整体框架设计

鉴于软件的总体功能需求是基于图形的绘制及应用, 本软件选用了开源跨平台的图形库 QT 平台进行开发。QT 系统基于使用面向对象的 C++ 语言开发, 提供了丰富的控件库以及优秀的 2D、3D 绘制功能^[2]。针对于大量 2D 图元绘制与管理的场景, QT 图形库提供了优秀的 Graphics View Framework 来应对这样的问题^[3]。

软件基于 QT 提供的 Graphics View 框架进行开发。QT 的 Graphics View 框架提供 View-Scene-Item 的三类对象, 供用户使用: Item 作为是所有图元的基类, 封装了图元公共属性; Scene 场景类, 管理所有的 Item; 并提供一个支持缩放与旋转的视图类 View 开对管理的图元进行显示。准确讲, QT 提供的这个框架只是一个基于 item 的 Model-View 框架, 为了方便将显示与控制逻辑进行解耦^[4], 参考经典的 MVC 架构加入了控制器类 GraphContext 类, Model 类继承至框架本身已有的 QGraphicsItem 类; Graph 类继承 QT 的 QGraphicsView 类; 加入了自定义的 Controller 类 GraphContext 类。三者的关系如图 1 所示^[5]。网络配线管理系统同其它绘图的 GUI 系统一样, 是事件驱动交互式用户界面系统, 而 GUI 的用户输入事件只有按键与鼠标两类事件。用户的输入事件经由视图类 Graph 捕捉到后, 交由控制器类 GraphContext 类进行处理。GraphContext 控制器类, 负责维护当前绘图的状态, 并根据当前绘图系统状态 (这也是类名取名 context 的原因), 决定如何响应该用户事件。例如, 同是鼠标点击事件, 在连线绘制时, 触发的是连线的绘制功能, 而在正常的浏览状态下, 鼠标移动则是执行的图元选择功能。

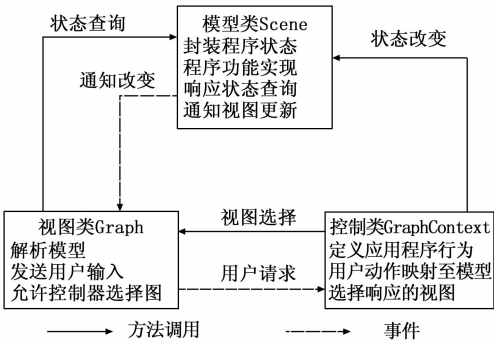


图 1 MVC 组件类型及功能

GraphContext 类作为系统的核心类, 其需要管理的一张绘图系统的方方面面, 为从功能上解耦系统的各个模块, 将系统功能细分为如图 2 所示的几大类功能模块, 每个模块功能实

现由专门的模块类予以实现。再由 GraphContext 将各个模块的功能组合在一起完成复杂的绘图系统。

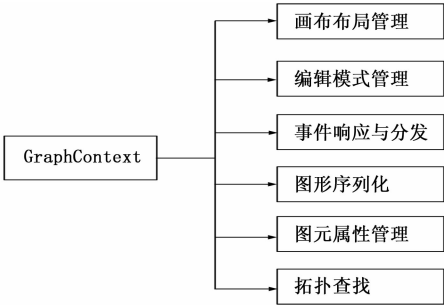


图 2 GraphContext 类的管理功能

2 关键功能实现

2.1 核心图元组件的构造

本软件面向机房配线管理, 主要涉及如服务器、工作站、交换机、路由器, 防火墙、机柜及其配线架等有关于网络连接的设备, 因此在编程中都将以上设备抽象为节点 (Node) 图元。节点内包含不定数量的网口 (Connector) 图元, 网口之间连接存在两种连接: 一种是模拟实际网线的连接 (Link); 另一种是配线架之间预连接的端口跳线 (ShortcutLink)。各类图元均继承于 QGraphicsItem 和 QGraphicsItem 这两个父类, 具体关系如图 3 所示。

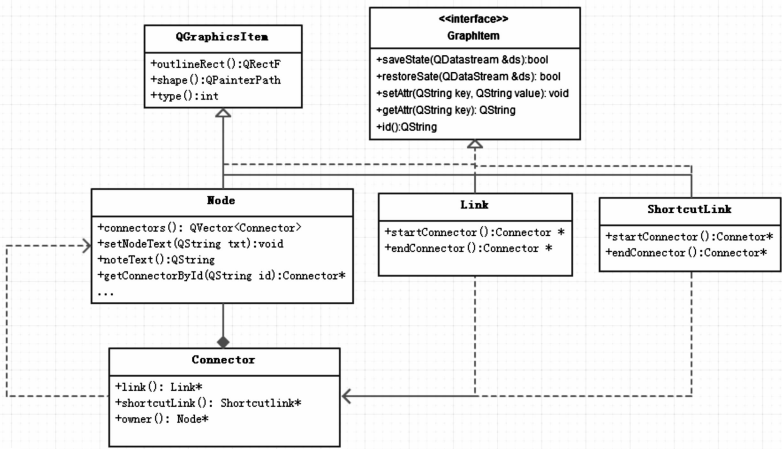


图 3 核心图元类构造及其关系

QGraphicsItem 类是 QT 视图框架的一部分, 是在一个 QGraphicsScene 中所有基本图形的父类, 它为绘制 item 提供了一个轻量级的窗口, 包括声明 item 的位置, 碰撞检测, 绘制重载和 item 之间的事件处理等。GraphItem 是自定义的基类, 用于包含所有类的基本属性和操作, 如所有图元的外边轮廓、保存和恢复状态、属性配置等接口。

Node 类是对现场设备的抽象, 在场景中以矩形作为外边框, 内含一定数量的“凸”形网口, 通过双击事件可以触发节点属性管理窗口。构造 Node 类时还需注意外形与画布网格对其关系, Node 文字标注与外形位置相对关系, Node 中心坐标与 Connector 坐标之间的关系等等。

Connector 是对设备网口的抽象, 附属在 Node 中存在的非独立图元。网口具有上下两种朝向, 网口之间可以通过

Link 和 ShortcutLink 两种图元连接;当网口捕捉到鼠标时,网口内的矩形热点会高亮;当网口位移变化时需要重新规划所连的接线或跳线。

Link 是对网线的抽象,两端依赖于 Connector 而存在。连线具有线型、粗细、颜色等基本属性同时,还具备所连端口、承载业务、所属屏柜等应用属性用于高级功能。连线具有端口跟踪和可编辑功能,用于节点端口位置变化后,具有虚线跟踪端口新位置,为用户更新连线提供线路指引。

ShortcutLink 与 Link 类似,但不需像 Link 根据画布的最小单元网格横平竖直走线,而是在两个网口之间以直线短接。在正常浏览、编辑状态下不显示短接线,这也是符合机房配线实际情形,因为短接线是机房建设预布置的固定走线,系统管理人员更多关注的是根据业务接入连线。

2.2 基于 XML 可配置自动生成的图元库

机房网络设备的网口数量不一,在图元设计时如果固定设备的网口数,将难以满足现场网口多样性的需求。本文采用 QT 提供的 QtXml 模块,调用 QDomStreamReader 解析 XML 配置文件自动生成图元库的方法^[6-7]。用户只需按标准的 XML 格式编写,预先定义 Node 图元的大小、名称、网口图元大小、端口数量及朝向、端口热点大小等图元基本参数,并存放到系统指定路径,当系统启动时,自动读取配置,生成相应的设备图元,在图元库悬浮窗口中显示。配置文件格式如下所示:

```
<? xml version=" 1. 0"? >
<cadrouting>
<entry term=" num" >
<property>1</property>
</entry>
<entry term=" kOutline" >
<entry term=" Width" >
<property>80</property>
</entry>
...
</cadrouting>
```

2.3 序列化与反序列化

网线配线管理系统不仅要提供图形绘制功能,还需要提供以文档形式保存与恢复图形的功能,这在软件工程上称之为序列化/反序列化^[8]。具体来说,序列化即是将画布上图元对象及其属性数据以特定格式存入磁盘空间,以文件的形式实现持久保存的过程。相对应地,反序列化即是系统载入图形文件后,重现图元对象、连接关系、属性记录等信息的过程。系统的序列化过程是根据 Item 类别遍历场景内的所有 Item 输出到磁盘文件。对于基本的 QT 自带的文字、矩形、直线等图元,其本身并没有与其它对象产生相互引用的关系,依赖 QT 提供的接口即可完成其序列化与反序列的。但对于自定义的 Node、Link 等,对象之间通过 C++ 指针建立起相互引用的树状关系。而指针只是对象在内存中的地址,如果直接将地址存入文件中,待反序列化恢复文件时,却不能保证新生成的文件依然在同一内存地址。显然,简单以保存内存地址的方式来序列化对象之间的引用关系行不通。为此,通过对系统中相互引用的对象,给其分配一个全局唯一 ID,保存对象之间引用关系时,存入文件中的不是其地址而是对象的 ID。同时,两两对象之间的引用关系转换成一 ID 存入一张新的引用关系表(叫做

ID 引用关系表)中。反序列化时,系统首先构造新对象;然后通过反序列化接口恢复其内部状态(其 ID 会恢复成序列化时存入文件的 ID);最后,当所有对象都恢复后,遍历 ID 引用关系表,恢复所有对象之间的引用关系。通过这种方式,即可以保证强关联的树状对象系统能得以正确的序列化与反序列化。在 ID 的生成方面,系统引入 UUID(Universally Unique Identifier)通用唯一标识^[9],其 16 个字节的长度足以保证 ID 的唯一性。

2.4 图形拓扑查找

调度自动化机房的网络接线比较复杂,一旦网络物理链路出现故障时,系统管理人员需根据已绘制的系统网络全图进行有针对性的排查。为提高故障诊断效率,本文开发了图形拓扑快速查找功能,即可通过选定某端口快速索引与其连接的所有端口及连线,或根据网络承载业务在网络全图中拓扑高亮标识。在绘制网络接线图时,端口之间通过连线建立了映射关系,应用拓扑算法实现指定端口连通的网络路径高亮显示,类似电力调度自动化系统中的电压拓扑着色功能^[10],从而便于系统管理人员索引、分析各根网线的连接情况和承载业务。

图形拓扑查找的关键是通过连线建立起来的“路由表”,查找的工作只是按照一定的算法在表中快速串联各图元对象,以按名字查找为例,伪代码实现如下:

```
FiterConnectorByName (string a) {
foreach (Connector * c, scene->connectors ()) {
//遍历场景中的所有网口
if (c. name. contains (a)) { //匹配端口名称
.....
}
}
```

2.5 属性管理及标签自动生成

将绘制图元增加应用属性是本软件的重要功能特性。除了图元的基本几何信息外,应用属性主要还包括设备位置、名称、端口 ID、对端端口 ID、端口 IP 等。图元属性数据需与画布中的图元动态关联,例如通过设置属性列表中图元名称的更改,可动态更新至画布中。

属性列表以悬浮窗口 RightDockWidget 类实现,该类继承于 QT 的“停靠”窗口 QDockWidget 父类,在窗口内创建 treeWidget 对象用于树形结构的属性内容视图。软件图元的双击事件触发到属性窗口显示,再利用 QT 的“信号与槽”机制实现属性值与图元数据的同步,即属性窗口检测到属性值变化后向绘图区域发送信号。例如:

```
connect (treeWidget, SIGNAL (itemChanged
(QTreeWidgetItem *, int)), RightDockWidget,
SLOT (propertyItemChanged (QTreeWidgetItem *,
int)));
```

标签管理一直是自动化机房标识管理的难点,存在包括线端数量多、描述格式不统一等问题。本软件在绘图功能的基础上结合图元属性管理,具有自动生成端口连接网线两端的标准标签功能。实现该功能的思路是遍历场景内的连接线,首先由连线找出两端网口编号 ID、设备所属机柜、承载业务,然后通过文本处理组织构成标签的起点、终点、业务三要素信息,调用 QT 的 QAxObject 类将标签输出生成 Excel 文档,最后由专业标签打印设备关联 Excel 文档,批量输出可粘贴的纸质标签。

(下转第 320 页)